

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-069191

(43)Date of publication of application : 03.03.2000

(51)Int.Cl.

H04M 11/00

G06F 3/00

G06F 13/00

(21)Application number : 10-234351

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 20.08.1998

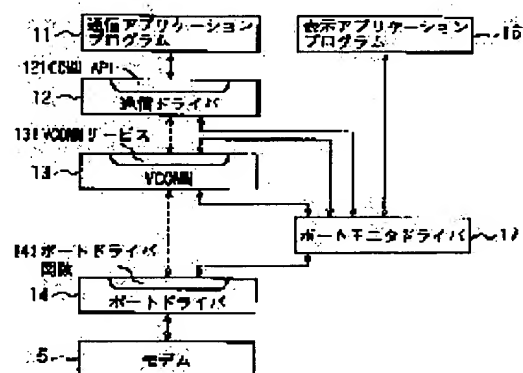
(72)Inventor : ISHIKAWA KATSUTOSHI

## (54) COMPUTER SYSTEM AND DATA COMMUNICATION INFORMATION DISPLAY METHOD

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a computer system capable of displaying a variety of data communication information such as the disconnection reason of a line, a connecting destination telephone number and charging information without accompanying the improvement of a communication application program.

**SOLUTION:** A communication driver 12 communicates with a VCOMM 13 by using a VCOMM service 131, the VCOMM 13 communicates with a port driver 14 by using a port driver function 141 and a port monitor driver 17 tentatively hooks (snatches or steals) the utilization of the VCOMM service 131 of the communication driver 12 and the utilization of the port driver function 141 of the VCOMM 13. Then, a display application program 16 displays the variety of the data communication information (disconnection reason of the line, the connecting destination telephone number and the charging information, etc.), obtained by the hook of the port monitor driver 17 to a user.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-69191

(P2000-69191A)

(43) 公開日 平成12年3月3日(2000.3.3)

(51) IntCl <sup>7</sup>	識別記号	F I	テマコード(参考)
H 0 4 M 11/00	3 0 2	H 0 4 M 11/00	3 0 2 5 B 0 8 9
G 0 6 F 3/00	6 5 2	G 0 6 F 3/00	6 5 2 A 5 K 1 0 1
13/00	3 5 3	13/00	3 5 3 U

審査請求 未請求 請求項の数10 O L (全 10 頁)

(21) 出願番号 特願平10-234351

(22) 出願日 平成10年8月20日(1998.8.20)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 石川 勝敏

東京都青梅市末広町2丁目9番地 株式会  
社東芝青梅工場内

(74) 代理人 100058479

弁理士 鈴江 武彦 (外6名)

Fターム(参考) 5B089 GA05 GA35 GB01 JB10 JB15

KA13 LB14

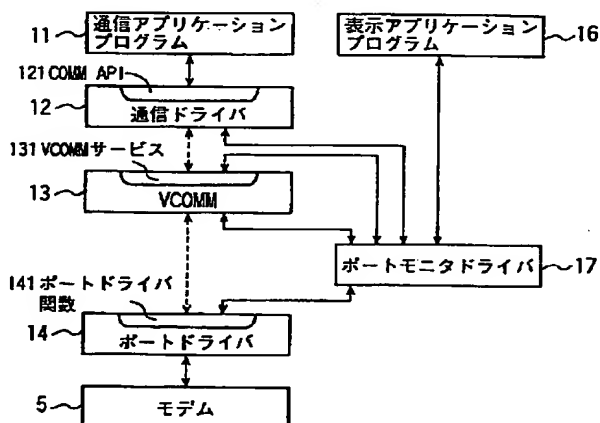
5K101 KK15 KK16 NN18

(54) 【発明の名称】 コンピュータシステムおよびデータ通信情報表示方法

(57) 【要約】

【課題】 回線の切断理由、接続先電話番号および課金情報などの各種データ通信情報の表示を通信アプリケーションプログラムの改良を伴わずに行なうことを可能とするコンピュータシステムを提供する。

【解決手段】 通信ドライバ12は、VCOMMサービス131を用いてVCOMM13と通信し、また、VCOMM13は、ポートドライバ関数141を用いてポートドライバ14と通信するものであり、ポートモニタドライバ17は、この通信ドライバ12のVCOMMサービス131の利用とVCOMM13のポートドライバ関数141の利用とを一時的にフック（横取り、あるいは盗み取り）する。そして、表示アプリケーションプログラム16は、このポートモニタドライバ17のフックにより得られる各種データ通信情報（回線の切断理由、接続先電話番号および課金情報など）をユーザに対して表示する。



## 【特許請求の範囲】

【請求項1】 回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、

前記デバイスに対するI/Oコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、

前記通信ポートを駆動制御するポートドライバと、

前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたI/Oコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、

前記I/Oコマンドに対する応答を前記通信制御手段より奪取するI/Oコマンドフック手段と、

前記I/Oコマンドフック手段が奪取した前記応答から所定のデータ通信情報を抽出するデータ通信情報抽出手段と、

前記データ通信情報抽出手段により抽出されたデータ通信情報を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。

【請求項2】 回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、

前記デバイスに対するI/Oコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、

前記通信ポートを駆動制御するポートドライバと、

前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたI/Oコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、

前記I/Oコマンドに対する応答を前記通信制御手段より奪取するI/Oコマンドフック手段と、

前記I/Oコマンドフック手段が奪取した前記応答から回線の切断理由を抽出するデータ通信情報抽出手段と、前記データ通信情報抽出手段により抽出された回線の切断理由を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。

【請求項3】 回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、

前記デバイスに対するI/Oコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、

前記通信ポートを駆動制御するポートドライバと、

前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたI/Oコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、

前記I/Oコマンドに対する応答を前記通信制御手段よ

り奪取するI/Oコマンドフック手段と、

前記I/Oコマンドフック手段が奪取した前記応答から接続先電話番号を抽出するデータ通信情報抽出手段と、前記データ通信情報抽出手段により抽出された接続先電話番号を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。

【請求項4】 回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、

前記デバイスに対するI/Oコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、

前記通信ポートを駆動制御するポートドライバと、

前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたI/Oコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、

前記I/Oコマンドに対する応答を前記通信制御手段より奪取するI/Oコマンドフック手段と、

前記I/Oコマンドフック手段が奪取した前記応答から課金情報を抽出するデータ通信情報抽出手段と、

前記データ通信情報抽出手段により抽出された課金情報を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。

【請求項5】 回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、

前記通信ポートを駆動制御するポートドライバと、

前記デバイスに対して発行されるI/Oコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、

前記I/Oコマンドに対する応答を前記通信制御手段より奪取するI/Oコマンドフック手段と、

前記I/Oコマンドフック手段が奪取した前記応答から所定のデータ通信情報を抽出して要求元に引き渡すデータ通信情報抽出手段とを具備することを特徴とするコンピュータシステム。

【請求項6】 回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、

前記デバイスに対して発行されるI/Oコマンドおよびその応答をその発行元と前記デバイスとの間で奪取するI/Oコマンドフック手段と、

前記I/Oコマンドフック手段が奪取した前記I/Oコマンドおよびその応答を本来の受け取り先に返却するI/Oコマンドリストア手段と、

前記I/Oコマンドフック手段が奪取した前記応答から所定のデータ通信情報を抽出して要求元に引き渡すデータ通信情報抽出手段とを具備することを特徴とするコンピュータシステム。

【請求項7】 回線網との接続に用いられるデバイスに

対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムのデータ通信情報表示方法であって、

前記デバイスに対して発行される I/O コマンドおよびその応答をその発行元と前記デバイスとの間で奪取するステップと、

前記奪取した前記応答の内容から所定のデータ通信情報を抽出するステップと、

前記抽出したデータ通信情報を表示するステップとを具備することを特徴とするデータ通信情報表示方法。

【請求項 8】 回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、

前記デバイスに対する I/O コマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、

前記通信ポートを駆動制御するポートドライバと、前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行された I/O コマンドに応じて前記ポートドライバを動作制御する通信制御手段と、

前記 I/O コマンドおよびその応答を前記通信制御手段より奪取する I/O コマンドフック手段と、

前記 I/O コマンドフック手段が奪取した前記 I/O コマンドおよびその応答を前記通信制御手段に返却する I/O コマンドリストア手段と、

前記 I/O コマンドフック手段が奪取した前記 I/O コマンドまたはその応答から回線の切断理由を抽出するデータ通信情報抽出手段と、

前記データ通信情報抽出手段により抽出された回線の切断理由を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。

【請求項 9】 回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、

前記デバイスに対する I/O コマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、

前記通信ポートを駆動制御するポートドライバと、前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行された I/O コマンドに応じて前記ポートドライバを動作制御する通信制御手段と、

前記 I/O コマンドおよびその応答を前記通信制御手段より奪取する I/O コマンドフック手段と、

前記 I/O コマンドフック手段が奪取した前記 I/O コマンドおよびその応答を前記通信制御手段に返却する I/O コマンドリストア手段と、

前記 I/O コマンドフック手段が奪取した前記 I/O コマンドまたはその応答から接続先電話番号を抽出するデ

ータ通信情報抽出手段と、

前記データ通信情報抽出手段により抽出された接続先電話番号を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。

【請求項 10】 回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、

前記デバイスに対する I/O コマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、

前記通信ポートを駆動制御するポートドライバと、前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行された I/O コマンドに応じて前記ポートドライバを動作制御する通信制御手段と、

前記 I/O コマンドおよびその応答を前記通信制御手段より奪取する I/O コマンドフック手段と、

前記 I/O コマンドフック手段が奪取した前記 I/O コマンドおよびその応答を前記通信制御手段に返却する I/O コマンドリストア手段と、

前記 I/O コマンドフック手段が奪取した前記 I/O コマンドおよびその応答から課金情報を抽出するデータ通信情報抽出手段と、

前記データ通信情報抽出手段により抽出された課金情報を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、たとえばモデムを用いたデータ通信を実行する通信アプリケーションプログラムが動作するコンピュータシステムおよび同システムのデータ通信情報表示方法に係り、特に、回線の切断理由、接続先電話番号および課金情報などの各種データ通信情報の表示を通信アプリケーションプログラムの改良を伴わずに行なうことを可能とするコンピュータシステムおよびデータ通信情報表示方法に関する。

【0002】

【従来の技術】 近年、デスクトップタイプやノートブックタイプなどと称される様々な種類の個人向けコンピュータ（パーソナルコンピュータ）が開発されている。また、この種のパーソナルコンピュータでは、インターネットや電子メールなどの普及に伴って、たとえばモデムなどといった、回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポート（あるいは、そのデバイスそのもの）を内蔵するのが一般的である。

【0003】 ところで、この種のパーソナルコンピュータ上で動作する、たとえばモデムを用いたデータ通信を実行する通信アプリケーションプログラム（インターネットブラウザソフトウェアや電子メールソフトウェアな

ど)は、たとえば文献「Windows95システムプログラム開発」(Walter Oney著、太田博志、小松克行訳、アスキー出版局出版、1997年10月11日 初版発行)などに記載された所定の手順でデータ通信を行なっているが、接続先電話番号や通信回線が切断された際の切断理由、あるいは課金情報などの各種データ通信情報を、モデムからのエコーバックやATコマンドなどから取得したり、その取得したデータ通信情報をユーザインタフェースとして表示するといったことは行なっていなかった。すなわち、たとえモデムがデータ通信終了時にATコマンドのエコーバックとして回線の切断理由を示すデータ通信情報を出力しても、通信アプリケーションプログラムはそれらを無視していた。

#### 【0004】

【発明が解決しようとする課題】この通信アプリケーションプログラムによるデータ通信情報の無視は、データ通信情報そのものが、安定度の高い有線回線を利用したデータ通信の実行時にはさほど必要視されていなかったことによる。しかしながら、最近では、携帯電話やPHSなどの無線回線を利用したデータ通信も盛んに行なわれるようになってきており、ユーザの意図しない回線の切断が大幅に増加している。そして、回線の切断理由を得られないユーザは、このような場合、そのほとんどが電波状況の悪化によって回線が切断されたと考え、改めて回線を接続しようとするが、電波状況の悪化によって回線が切断されたのではなく、接続先の着信拒否により回線が切断されたときには、ユーザに無駄な操作を強いることになってしまう。

【0005】この発明はこのような実情に鑑みてなされたものであり、回線の切断理由、接続先電話番号および課金情報などの各種データ通信情報の表示を通信アプリケーションプログラムの改良などを伴わずに行なうことを可能とするコンピュータシステムおよびデータ通信情報表示方法を提供することを目的とする。

#### 【0006】

【課題を解決するための手段】この発明は、前述した目的を達成するために、たとえばインターネットブラウザソフトウェアや電子メールソフトウェアなどがモデムなどに対して発行するI/Oコマンドおよびその応答を、その発行元とモデムとの間のいずれかで盗み取り、この盗み取ったI/Oコマンドおよびその応答から回線の切断理由、接続先電話番号および課金情報などの各種データ通信情報を抽出するとともに、その盗み取ったI/Oコマンドおよびその応答を、本来の受け取り先に何もなかったかのように戻すようにしたものである。

【0007】この発明においては、従来のシステム資源にI/Oコマンドおよびその応答の奪取をなんら認識させることがないため、通信アプリケーションプログラムの改良などを伴わずに各種データ通信情報を表示する

ことを可能とし、たとえば回線の切断理由を表示すれば、接続先の着信拒否により回線が切断されたときなど、ユーザに無駄な操作を強いることを防止する。

【0008】また、回線の接続時に接続先電話番号を抽出して表示すれば、ユーザに接続先の電話番号が正しいかどうか確認させる機会を与えることができ、誤った電話番号を指定して回線の接続を実施していたときなど、その誤りをより早く認識させることが可能となる。

【0009】さらに、回線の切断時に課金情報を抽出して表示すれば、たとえばシステム本体に備えられたタイマ機能により回線接続時間を計時して課金額を算出するといった複雑かつ誤差を生じ易い処理を行なわずとも、容易かつ正確に課金額をユーザに提供することが可能となる。

#### 【0010】

【発明の実施の形態】以下、図面を参照してこの発明の実施形態を説明する。図1は、この発明の実施形態に係るコンピュータシステムの概略機器構成を示す図である。この実施形態のコンピュータシステムは、たとえばデスクトップやノートブックタイプのパーソナルコンピュータなどであり、図1に示すように、CPU1、システムメモリ2、ディスプレイコントローラ3、磁気ディスク装置(HDD)4、モデム5およびキーボードコントローラ6を備えている。

【0011】CPU1は、このコンピュータシステム全体の制御を司るものであり、システムメモリ2に格納されたオペレーティングシステムやハードウェアドライバ、ユーティリティを含むアプリケーションプログラムなどを実行制御する。

【0012】システムメモリ2は、このコンピュータシステムの主記憶装置となるメモリデバイスであり、CPU1によって実行制御されるオペレーティングシステムやハードウェアドライバ、ユーティリティを含むアプリケーションプログラム、およびこれらの実行に用いられる各種処理データなどを格納する。

【0013】ディスプレイコントローラ3は、ユーザインタフェースのアウトプットを司るデバイスであり、CPU1により描画されたビデオメモリ(VRAM)上の表示データをLCDなどのディスプレイに表示する。

【0014】磁気ディスク装置(HDD)4は、このコンピュータシステムの補助記憶装置となる大容量のメモリデバイスであり、システムメモリ2にロードされるプログラムや処理データ、あるいはシステムメモリ2から退避されるプログラムや処理データなどを格納する。

【0015】モデム5は、回線網(アナログ)との接続に用いられるデバイスであり、回線網を介させて他のコンピュータシステムとの間で送受信されるデータをアナログ/デジタル変換する。

【0016】そして、キーボードコントローラ6は、ユーザインタフェースの入力を司るデバイスであ

り、キーボードやマウスから送られる操作内容を示す制御信号を受け取ってCPU1に転送する。

【0017】図2には、このような機器構成をもつこの実施形態のコンピュータシステムの機能ブロックが示されている。なお、この図2では、米国マイクロソフト社により開発されたWindows95オペレーティングシステム下での機能ブロックを示すが、この発明は、これに限定されるものではない。すなわち、オペレーティングシステムの形式に依存して変形させることは、この発明の範囲から逸脱することなしに容易に可能である。

【0018】この実施形態のコンピュータシステムでは、モデム5へのアクセスを必要とする一つまたはそれ以上の通信アプリケーションプログラム11（リモートネットワークアクセス、ファックスアプリケーション、電子メールアプリケーションおよび掲示板サービスアプリケーションなど）に対するサポートを提供する。この通信アプリケーションプログラム11は、モデム5へのアクセスを必要とするものであり、そのモデム5に対するI/Oコマンドの発行によりデータI/Oを実行する。

【0019】通信ドライバ12は、COMM API121をサポートし、通信アプリケーションプログラム11に装置に依存しない方法でモデム5を使用することを許可する。この通信ドライバ12は、通信アプリケーションプログラム11が通信デバイスをオープンし、通信デバイスから読み出し、通信デバイスに書込むことを許可する機能を含む。また、COMM API121のデータI/O機能呼び出し、最終的にモデム5を通じてデータを送受信する。そして、ユーザは、通信アプリケーションプログラム11からユーザインタフェース機能を通じてモデム5の動作特性選択し、コンフィギュレーション機能を通じてモデム5の動作特性を決定または変更することができる。

【0020】VCOMM13は、いわゆる仮想デバイスであり、システムにおけるシリアルポート、パラレルポートおよびモデムなどといった通信資源へのすべてのアクセスを管理する。そして、通信ドライバ12は、このVCOMM13によって供給されるVCOMMサービス131によってVCOMM13と通信する。

【0021】ポートドライバ14は、デバイス特定のコードを含んでおり、これにより通信アプリケーションプログラム11がポートに接続された特定のデバイスと通信することを可能にする。

【0022】そして、この実施形態のコンピュータシステムは、ポートモニタドライバ17と表示アプリケーションプログラム16とを備えた点を特徴とするものであり、ポートモニタドライバ17は、VCOMM13とポートドライバ14との間の通信の監視、通信データの解析および通信データの変更などを可能とするものである。なお、図2は、通信アプリケーションプログラム1

1とモデム5との間の通信の監視や解析、変更を施すための一例を示すものであり、この発明はこの構成に限定されるものではない。たとえば、ポートモニタドライバ17は、通信ドライバ12とVCOMM13との間の通信を監視、解析または変更することも可能である。

【0023】従来、通信ドライバ12は、VCOMM13によって供給されるVCOMMサービス131を用いてVCOMM13と通信し、また、VCOMM13は、ポートドライバ14によって供給されるポートドライバ関数141を用いてポートドライバ14と通信するが、図2に示すこの実施形態においては、ポートモニタドライバ17が通信ドライバ12のVCOMMサービス131の利用をフック（横取り、あるいは盗み取り）したり、また、VCOMM13のポートドライバ関数141の利用をフックすることにより、実質的に、通信ドライバ12はポートモニタドライバ17と通信し、また、ポートモニタドライバ17はVCOMM12と通信することになり、さらに、VCOMM13はポートモニタドライバ17と通信し、また、ポートモニタドライバ17はポートドライバ14と通信することになる。

【0024】そして、表示アプリケーションプログラム16は、ポートモニタドライバ17が通信ドライバ12のVCOMMサービス131の利用やVCOMM13のポートドライバ関数141の利用をフックすることによって得られる各種データ通信情報（回線の切断理由、接続先電話番号および課金情報など）をポップアップ画面などによりユーザに対して表示するものである。

【0025】図3は、この実施形態のコンピュータシステムにおけるシステムメモリ2上の構成例を示すものである。VCOMM13は、VCOMMサービス131のエントリをシステムメモリ2上に確保している。図3には、\_\_VCOMM\_OpenComm1311、\_\_VCOMM\_Add\_Port1312、\_\_VCOMM\_CloseComm1313などのVCOMMサービス131の例が示されている。通信ドライバ12は、通信アプリケーションプログラム11からのCOMM API121を利用した命令に基づき、\_\_VCOMM\_OpenComm1311、\_\_VCOMM\_Add\_Port1312などのVCOMMサービス131を利用して、通信ポートのオープンのための準備やポートの管理機構の構築などを行ない、最後に\_\_VCOMM\_CloseComm1313を用いてポートをクローズする。

【0026】ポートドライバ14は、オープンされたポートを管理するためのPortData構造体141、ポートをオープンするためのPortOpen関数ポインタ142などをメモリ上に確保している。また、PortData構造体141には、構造体のサイズ情報1411、構造体のバージョン情報1412、ポートI/Oを行なうための様々な関数へのポインタ群143へのポインタ1413、ポインタ群143に含まれる関数の

数情報1414などが要素として含まれ、ポートI/O関数ポインタ群143には、ポートをクローズするための関数PortCloseへのポインタ1431、ポートからデータを読み出すための関数PortReadへのポインタ1432、ポートヘデータを書き出すための関数PortWriteへのポインタ1433などが含まれる。

【0027】ポートモニタドライバ17は、ポートをオープンするための関数であるPortOpenへのポインタ142をコピーして格納する領域OpenFunc1711、ポートI/O関数ポインタ群143へのポインタをコピーして格納するOriginalFunc1712、ポートモニタドライバ17独自のポートI/O関数ポインタ群172へのポインタを格納するLocalFunc1713などを持った構造体OpenInfo171を含んでいる。ポートモニタドライバ17独自のポートI/O関数群173には、たとえばポートをオープンするための準備を行なうLocalAddPort1731、ポートをオープンするLocalOpen1732、ポートをクローズするLocalClose1733、ポートからデータを読み出すLocalRead1734、ポートヘデータを書き出すLocalWrite1735などがある。また、ポートモニタドライバ17独自のポートI/O関数ポインタ群172には、LocalClose1733へのポインタLocalPortClose1721、LocalRead1734へのポインタLocalPortRead1722、LocalWrite1735へのポインタLocalPortWrite1723などが含まれている。

【0028】また、ライトバッファ174およびリードバッファ175は、それぞれポートへ書き出すデータを一時格納するバッファ、ポートから読み出したデータを一時格納するバッファである。

【0029】さらに、ポートモニタドライバ17は、表示アプリケーションプログラム16に情報を通知するため、表示アプリケーションプログラム16が持つ関数へのポインタ群CallbackFunc構造体176を含んでいる。図3は、表示アプリケーションプログラム176が持つ3種類の関数へのポインタをCallbackFunc構造体176が含んでいる例であり、CallbackFunc1(1761)、CallbackFunc2(1762)、CallbackFunc3(1763)がそれである。

【0030】以下、図4乃至図12のフローチャートを参照してこのコンピュータシステムの動作原理を説明する。図4は、ポートモニタドライバ17の初期化を示すフローチャートである。たとえば表示アプリケーションプログラム16などの上位アプリケーションによりポートモニタドライバ17がロードされると、ポートモニタ

ドライバ17は、最初に初期化作業として、VCOMMサービス131のうちの\_VCOMM\_Add\_Port1312のフックを行なう(ステップA1)。従来、通信ドライバ12がVCOMMサービス131のうちの\_VCOMM\_Add\_Port1312をコールした際には、VCOMM13内に用意されている関数がコールされるが、その関数へのポインタが登録されている部分をポートモニタドライバ17内のLocalAddPort1731と置き換えることにより、LocalAddPort1731で示されるポートモニタドライバ17内のローカル関数がコールされるようになる。そして、ポートモニタドライバ17は、そのローカル関数内で独自の処理を行なった後、元の本来の関数をコールし直すことにより、フックしたこととなる。次に、ポートモニタドライバ17は、OpenInfo構造体171の初期化を行なう(ステップA2)。これは、たとえば構造体の各要素に0(ゼロ)を代入して初期化するという、一般的なメモリ初期化作業である。

【0031】図5は、CallbackFunc構造体176へのコールバック関数の登録を示すフローチャートである。たとえば表示アプリケーションプログラム16などの上位アプリケーションによりポートモニタドライバ17がロードされる際、表示アプリケーションプログラム16は、自身に含まれる関数のポインタをポートモニタドライバ17へ通知することにより、その関数のポインタをCallbackFunc構造体176へ登録する(ステップB1)。この処理により、ポートモニタドライバ17は、情報を表示アプリケーションプログラム16に通知するためにどの関数(ルーチン)をコールすれば良いか知ることができる。

【0032】図6は、通信ドライバ12がVCOMMサービス131のうちの\_VCOMM\_Add\_Port1312をコールした際、図4のステップA1の処理を行なったことによりポートモニタドライバ17内のローカル関数LocalAddPort1731が呼ばれたときに、そのローカル関数LocalAddPort1731が実際に実行する処理を示すフローチャートである。\_VCOMM\_Add\_Portコールにより呼ばれるこの関数は、この実施形態のWindows95のオペレーティングシステムの場合、ポートを実際にオープンする際にコールされる関数のポインタ(この場合はPortOpen142)を得られる仕様となっているため、このLocalAddPort1731においても同様にPortOpen142を得て、それをOpenInfo構造体171のOpenFunc1711へ格納する(ステップC1)。これにより、OpenFunc1711ポインタは、PortOpen142を指し示すこととなる(矢印a)。次に、この関数は、フックしなかった場合に\_VCOMM\_Add\_Portコールによりコールされるはずであった本来の関数をコー



ルする(ステップC2)。このとき、ポートモニタドライバ17が独自に持つオープン関数へのポインタLocalOpen1732をアーギュメントとして渡すことにより、それ以後、ポートを実際にオープンする際には、本来の関数PortOpen142がコールされる代わりに、ローカル関数LocalOpen1732がコールされることになる。

【0033】図7は、図6のステップC2の処理を行なったことにより、ポートをオープンする際にポートモニタドライバ17内のローカル関数LocalOpen1732がコールされたときに、そのローカル関数LocalOpen1732が実際に実行する処理を示すフローチャートである。この関数は、まず始めに、図6のステップC1の処理によりOpenFunc1711に格納した本来のPortOpen142をコールする(ステップD1)。Windows95のオペレーティングシステムにおいては、本来のPortOpen142を返り値としてPortData構造体141へのポインタが得られるので、その要素よりPortFunctions1413を取得し、ポートI/O関数ポインタ群143へのポインタを得て、それをOriginalFunc1712へ格納する(ステップD2)。さらに、ポートモニタドライバ17が持つ独自のポートI/O関数ポインタ群172内に、本来のポートI/O関数ポインタ群143の内容をコピーした後(ステップD3)、ポートモニタドライバ17にてフックする機能(関数)のみ、独自のポートI/O関数ポインタ群172の内容を変更する(ステップD4)。図3ではは、一例として、ポートをクローズする際にコールされるPortClose1431、ポートからデータを読み出すためのPortRead1432、ポートへデータを書き出すためのPortWrite1433をフックするために、独自のポートI/O関数ポインタ群172の要素のうち、それら三つの関数のポインタをそれぞれLocalPortClose1721、LocalPortRead1722、LocalPortWrite1723に変更している。LocalPortClose1721は、ポートモニタドライバ17が独自に持つ関数LocalClose1733、LocalPortRead1722は、ポートモニタドライバ17が独自に持つ関数LocalRead1734、LocalPortWrite1723は、ポートモニタドライバ17が独自に持つ関数LocalWrite1735を指し示す。そして、すでに得ているPortData構造体141に含まれるPortFunctions1413をポートモニタドライバ17独自のポートI/O関数ポインタ群172へのポインタと置き換える(ステップD5)。こうすることにより、それ以降、VCOMM13がオペレーティングシステムの動作に基づいてポートドライバ14に含まれる関数をコールする際、コールする

関数へのポインタを含んでいるPortData構造体141のPortFunctions1413がポートモニタドライバ17所有のポートI/O関数ポインタ群171を指しているため、ポートモニタドライバ17内のローカル関数LocalClose1733、LocalRead1734およびLocalWrite1735などがコールされることになる。

【0034】図8は、ポートモニタドライバ17内のローカル関数LocalRead1734がコールされた際の処理を示すフローチャートである。通信アプリケーションプログラム11からポートリード要求があった際、図3乃至図7に示した処理によりPortRead1432のコールがフックされ、LocalRead1734がコールされる。この関数では、まず始めに、OpenInfo構造体171に含まれるOriginalFunc1712より、本来のポートI/O関数ポインタ群143を得て、さらにそこから得られるPortRead1432をコールすることにより、本来のポートリード処理を行なう(ステップE1)。そして、ポートリード処理より得られた、たとえばモデムの出力などのリードデータを解析し(ステップE2)、電話通信切断時の回線切断理由に関わる情報や電話通信終了時の課金に関わる情報などを取得する。

【0035】図9は、ポートモニタドライバ17内のローカル関数LocalWrite1735がコールされた際の処理を示すフローチャートである。通信アプリケーションプログラム11からポートライト要求があった際、図3乃至図7に示した処理によりPortWrite1433のコールがフックされ、LocalWrite1735がコールされる。この関数では、まず始めに、OpenInfo構造体171に含まれるOriginalFunc1713より、本来のポートI/O関数ポインタ群143を得て、さらにそこから得られるPortWrite1433をコールすることにより、本来のポートライト処理を行なう(ステップF1)。そして、ポートライト要求から得られた、たとえば通信アプリケーションプログラム11からモデムへのATコマンド書き出しなどのライトデータを解析し(ステップF2)、接続先の電話番号などの情報を取得する。

【0036】図10は、ポートモニタドライバ17内のローカル関数LocalClose1733がコールされた際の処理を示すフローチャートである。通信アプリケーションプログラム11からポートクローズ要求があった際、図3乃至図6に示した処理によりPortClose1431のコールがフックされ、LocalClose1733がコールされる。この関数では、まず始めに、OpenInfo構造体171に含まれるOriginalFunc1713より、本来のポートI/O関数ポインタ群143を得て、さらにそこから得られるPortClose1431をコールすることにより、

本来のポートクローズ処理を行なう(ステップG1)。そして、OpenInfo構造体171を初期化する(ステップG2)。

【0037】図11は、図8のステップE2のリードデータ解析処理の詳細を示すフローチャートである。図8のステップE1の処理によりポートからのリードデータが得られた際、まず、リードデータをリードバッファ175にコピーする(ステップH1)。たとえば通信アプリケーションプログラム11とモデム5との間でI/OされるATコマンドのリザルトコード文字列を解析する場合には、CR(キャリッジリターン)コードがひとつのリザルトコードの区切りになるので、リードバッファ175にはCRコードまでを一時格納する。そして、たとえば通信が終了または切断された場合にモデム5から出力されるリザルトコード「NO CARRIER」を検出するため、リードバッファ175に格納された文字列のうちの最初の10文字を「NO CARRIER」文字列と比較し(ステップH2)、一致した場合には(ステップH2の“一致”)、CallbackFunc構造体176に格納される任意の関数、たとえばCallbackFunc1をコールし、リードバッファ175の内容を表示アプリケーションプログラム16に通知する(ステップH3)。一方、表示アプリケーションプログラム16は、たとえば得られたリードバッファ175の文字列に「NO CARRIER:RECEIVED REFUSED」が格納されている場合には、「:RECEIVED」文字列以降が回線切断の理由として、「REFUSED」を切り出し、図13に示すように画面上に回線切断理由を表示する。この例は、電話相手先が着信拒否をしたために回線が切断されたことを意味している。

【0038】また、たとえば課金情報を表示する場合には、通信が終了した場合にモデム5から出力されるリザルトコード「CHARGE=」を検出してCallbackFunc構造体176に格納される任意の関数をコールすることにより、リードバッファ175の内容を表示アプリケーションプログラム16に通知する。この場合の文字列は、たとえば通信料金が100円のときは「CHARGE=100」、通信料金が1270円のときは「CHARGE=1270」のように構成されるので、表示アプリケーションプログラム16は、「100」や「1270」を切り出して、課金情報として画面上に表示すればよい。

【0039】図12は、図9のステップF2のライトデータ解析処理の詳細を示すフローチャートである。図9のステップF1の処理により通信アプリケーション11からモデム5へのライトデータが得られた際、まず、ライトデータをライトバッファ174にコピーする(ステップI1)。たとえば通信アプリケーションプログラム11とモデム5との間でI/OされるATコマンドを解

析する場合には、CR(キャリッジリターン)コードがひとつのATコマンドの区切りになるので、ライトバッファ174にはCRコードまでを一時格納する。そして、たとえば通信アプリケーションプログラム11がATコマンドを用いて電話発呼処理を行なう際の接続先電話番号を得るべく「ATD」(発呼を行うためのATコマンド)という3文字の文字列を検出するため、ライトバッファ174に格納された文字列のうちの最初の3文字を「ATD」文字列と比較し(ステップI2)、一致した場合には(ステップI2の“一致”)、たとえば「ATD03-777-7777」のようにATコマンド文字列に接続先電話番号が続いているので、ATD以降の数字および括弧やハイフンなどの文字列を一時的なバッファに格納する(ステップI3～ステップI4)。そして、一連の電話番号が得られてから、CallbackFunc構造体176に格納される任意の関数、たとえばCallbackFunc2をコールし、ライトバッファ174の内容を表示アプリケーションプログラム16に通知する(ステップI5)。一方、表示アプリケーションプログラム16は、たとえば得られた接続先電話番号の情報を図13のように画面上に表示する。

【0040】このように、この実施形態のコンピュータシステムにおいては、通信アプリケーションプログラム11がモデム5の出力する切断理由、接続先電話番号および課金情報などの各種データ通信情報を無視するものであっても、この通信アプリケーションプログラム11とは別途独立して、すなわち、この通信アプリケーションプログラム11の改良を伴うことなく、ユーザにとって有意義な情報を表示することが可能となる。

【0041】

【発明の効果】以上詳述したように、この発明においては、コンピュータシステム上の任意のデータ通信アプリケーションプログラムがモデムなどを介してデータ通信を行なう際に、そのデータ通信アプリケーションプログラムからモデムに対して発行されるリードやライトなどのI/Oコマンドを盗み取り、この盗み取ったI/Oコマンドから回線の切断理由、接続先電話番号および課金情報などの各種データ通信情報を抽出するとともに、その盗み取ったI/Oコマンドを本来の受け取り先に何もなかったかのように返却するため、通信アプリケーションプログラムの改良などを伴わずに、この抽出した各種データ通信情報をユーザに知らせることが可能となる。

【図面の簡単な説明】

【図1】この発明の実施形態に係るコンピュータシステムの概略機器構成を示す図。

【図2】同実施形態のコンピュータシステムの機能ブロックを示す図。

【図3】同実施形態のコンピュータシステムにおけるシステムメモリ上の構成例を示す図。

【図4】同実施形態のポートモニタドライバの初期化を示すフローチャート。

【図5】同実施形態のCallbackFunc構造体へのコールバック関数の登録を示すフローチャート。

【図6】同実施形態の通信ドライバがVCOMMサービスのうちの\_VCOMM\_Add\_Port1をコールした際に呼ばれるポートモニタドライバ内のローカル関数LocalAddPortが実行する処理を示すフローチャート。

【図7】同実施形態のポートをオープンする際にコールされるポートモニタドライバ内のローカル関数LocalOpenが実行する処理を示すフローチャート。

【図8】同実施形態のポートモニタドライバ内のローカル関数LocalReadがコールされた際の処理を示すフローチャート。

【図9】同実施形態のポートモニタドライバ内のローカル関数LocalWriteがコールされた際の処理を示すフローチャート。

【図10】同実施形態のポートモニタドライバ内のローカル関数LocalCloseがコールされた際の処理を示すフローチャート。

【図11】同実施形態のポートモニタドライバ内のローカル関数LocalRead中のリードデータ解析処理の詳細を示すフローチャート。

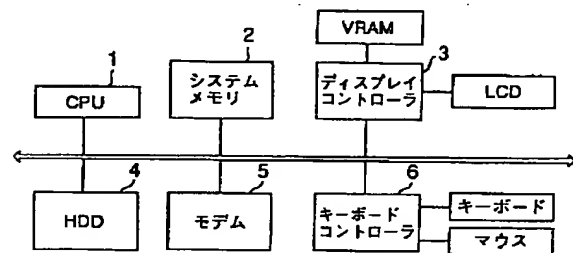
【図12】同実施形態のポートモニタドライバ内のローカル関数LocalWrite中のライトデータ解析処理の詳細を示すフローチャート。

【図13】同実施形態の表示アプリケーションプログラムによる各種データ通信情報の表示例を示す図。

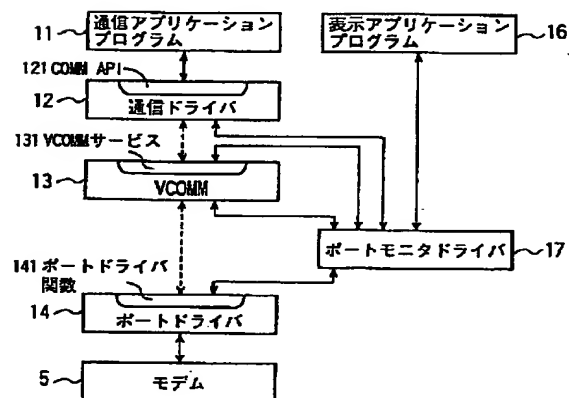
【符号の説明】

1…CPU、2…システムメモリ、3…ディスプレイコントローラ、4…磁気ディスク装置（HDD）、5…モデム、6…キーボードコントローラ、11…通信アプリケーションプログラム、12…通信ドライバ、13…VCOMMサービス、14…ポートドライバ、16…表示アプリケーションプログラム、17…ポートモニタドライバ。

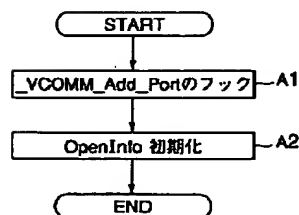
【図1】



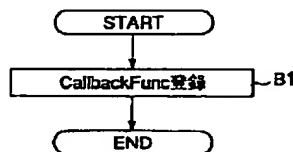
【図2】



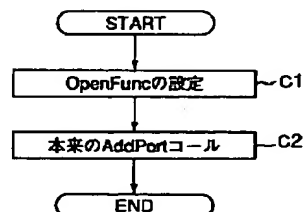
【図4】



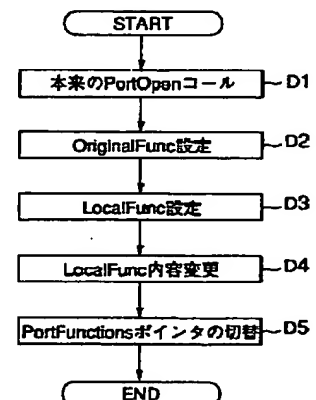
【図5】



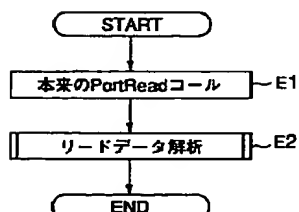
【図6】



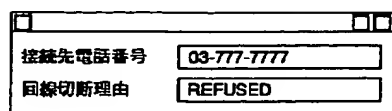
【図7】



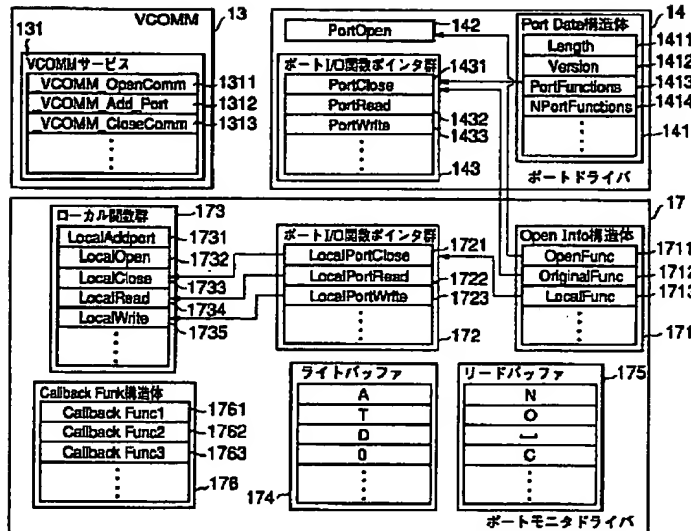
【図8】



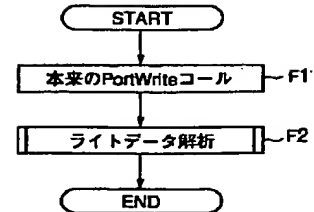
【図13】



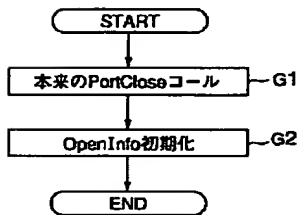
【図3】



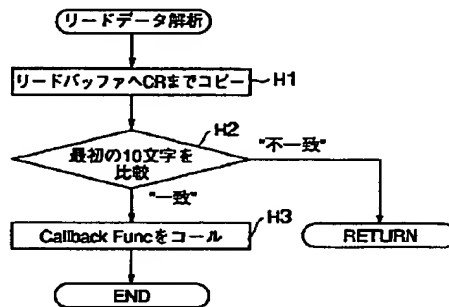
【図9】



【図10】



【図11】



【図12】

